COMMANDS

COMMAND CATEGORIES

--> CMDS TO GET AROUND

--> CMDS TO LOOK/EDIT

--> CMDS TO SEARCH

--> CMDS FOR EASE

---

cd -

- cd will change directory, with the addition of `-` it will go back to the last directory you were at. This includes descending to a deep directory with a long path.
  - e.g. starting at /documents, if you run:
    - `cd folder1/folder2/folder3' to get to to folder three.
    - instead of running something like `cd ../../../` to get back to /documents you can run `cd -` inside /folder3 to get back.

cd --

- This will get you back to your home directory.
- Alternatively the command `cd ~` will do the same.

cat

- If you want to "peek" into a file you can use `cat fileName.js`.
  - This will output the contents of "fileName.js" to the terminal.
- To quickly see package.json files dependancies from the command line: `cat package.json' --> prints the contents to terminal. No need to open your text editor.

nano

- This is a very simple terminal based text editor. You can use this to add some quick comments or code.
- An alternative to "nano" is "VIM", a very popular terminal editor. This is more complex to learn but once learned it can be very efficient to code with.
- To open a file to edit with nano run, `nano existingFile.js`
- To create & edit a file run, `nano nonExistantFileRightNow.js' --> upon save this file will be created with your contents you added.

> *and* >>

- These commands can be used to write to a file, either overwriting the existing content and replacing it (>) or appending to the end (>>)
- Want to create and start readme.md , `cat > 'README.md`--> the console will drop a line and you can type what ever you want, using "ctr-c" to exit
- likewise if you want to append to the end of a file, `cat >> existingFile.txt`
- Don't forget you could also just `nano` into the file to edit. **(>) is destructive to any existing file you run it on**!

pbcopy

- Great way to copy text to your clipboard.
  - e.g. `pbcopy < fileToCopy.txt` --> entire file copied to clipboard
  - e.g. `grep '//' someCoolCommentsToShare.js | pbcopy` --> This will use grep to search a file and output what it finds through the pipes and into pbcopy command. you now have that in your clipboard.

history

- Output history of commands you have run up to a cap. Run `echo $HISTSIZE`
- Search for history of commands you have run with second arg. e.g. `history cd` --> this will find
- Once you have history output you can run a cmd directly with: `!<number-of-history-command>`

grep

- This is the command to run regular expressions on files or output using pipes.

man

- When given an argument it will output a manual about how to use that command.
- e.g. `man ls` --> This will output a manual that tells you about the ls command.
- This is a great resource you have right in the terminal to tell you how things work. This is also good in conjunction with the `--help` flag.
  - e.g. `ls --help` -->
- Output can be long so use return key to progress line my line, and spacebar to skip a full terminal screen.

!!

- Unity is the harmony of all visual elements in a composition. No single element takes over.

^mistake^correction

- Unity is the harmony of all visual elements in a composition. No single element takes over.

&&

- what if you want to run a command and then another one and you dont want to wait for the first to finish before you run the next...
- using && between commands will order them. first command with run and then when complete the next will run.
- e.g. `npm i && npm start` --> in one line this will install and then run your start command of your application. you can chain multiple commands using `&&`

|

- Piping is useful in the terminal to "chaining" or "pipe" outputs from a command into another command.
- e.g. `history | grep 'docker'` --> this command will run the history command and "pipe" the output to grep. Now grep will have that output to search through to match the string 'docker'. This would be good to seclude all your past docker commands.

cal and date

- Calendar and Date. Both respectively will print the current month with highlighted current day, and the current date and time (24hr).
- Look more into these for yourself. These will be a quick way to see the current time.

Reference:

Here is a helpful link to go over A LOT of commands. (https://ss64.com/osx/)